

A Degenerate Agglomerative Hierarchical Clustering Algorithm for Community Detection

Antonio Maria Fiscarelli¹, Aleksandr Beliakov, Stanislav Konchenko, and
Pascal Bouvry²

¹ C2DH, CSC-ILIAL, University of Luxembourg, 2, avenue de l'Université, 4365
Esch-sur-Alzette, Luxembourg

`antonio.fiscarelli@uni.lu`,

`https://www.c2dh.uni.lu/people/antonio-fiscarelli.0.0`

² FSTC-CSC/ILIAS & SnT, University of Luxembourg, 2, avenue de l'Université,
4365 Esch-sur-Alzette, Luxembourg

`pascal.bouvry@uni.lu`,

`http://staff.uni.lu/pascal.bouvry`

Abstract. Community detection consists of grouping related vertices that usually show high intra-cluster connectivity and low inter-cluster connectivity. This is an important feature that many networks exhibit and detecting such communities can be challenging, especially when they are densely connected. The method we propose is a degenerate agglomerative hierarchical clustering algorithm (DAHCA) that aims at finding a community structure in networks. We tested this method using common classes of graph benchmarks and compared it to some state-of-the-art community detection algorithms.

Keywords: Community detection, Graph clustering, Graph theory.

1 Introduction

Many complex systems such as social networks [1], the world wide web [2] and biological networks [3] can be represented using graphs. One of their many properties is the organisation into communities. Often different communities merge and form a hierarchical structure. Also, they differ in size and vertices show different degrees of connectivity. The community structure of a network can give access to relevant information about the dynamics of the network and its characteristics, this is why this has become a very relevant topic in computer science and other disciplines.

The method we propose is a degenerate agglomerative hierarchical clustering algorithm (DAHCA) that aims at finding community structures in networks. We have investigated how effectively our method can detect nested communities and discovered that it can detect both community and subcommunity structure. Next, we have compared our method to some of the state-of-the-art algorithms on the Girvan-Newman benchmark [4] and discovered that it can effectively detect communities and in some cases outperform other algorithms. Finally, our

method has been tested on the Zachary karate club network [5], a well known real-world network used as a benchmark for community detection algorithms.

2 Community detection: Problem definition and related work

This section presents some of the existing community detection algorithms. Community detection consists of grouping related vertices that usually show high intra-cluster connectivity and low inter-cluster connectivity. Many different methods have been proposed over the last years and contributions came from disciplines such as computer science, applied mathematics, physics, biology, economics and so on. However there is no best algorithm. Some algorithms simply perform better or are faster for different types of networks or different applications.

The method proposed by Newman and Girvan in [6] extends the definition of betweenness centrality to edges. Edges connecting communities will have a high edge betweenness and removing them will enhance the community structure of the network (BETW). On the other hand, Clauset, Newman and Moore [7] use a modularity measure in order to define communities that have many edges within them and few between them (GREEDY). Furthermore, Raghavan, Albert and Kumara [8] use a decentralised technique based on the majority rule to assign vertices to clusters (LAB PROP). The method that Pons and Latapy [9] describe uses random walks in order to define communities. Generally, random walkers tend to stay more in the same community (TRAP). Rosvall and Bergstrom [10] approach the problem using an information theoretic point of view to discover communities by using the probability flow of random walks (INFOMAP). Finally, the method proposed by Newman [11] is based on the eigenspectrum of the modularity matrix in order to maximize the modularity measure (EIGEN). The most used metric [12] [13] to evaluate community detection algorithms is the Normalized Mutual Information (NMI): it measures the agreement between communities and clusters found by a community detection algorithm [14]. $NMI = 1$ corresponds to perfect assignments, while $NMI = 0$ corresponds to completely independent assignments. Completeness (COMP) measures how vertices of a community are assigned to the same cluster, while homogeneity (HOMOG) measures how every cluster contains only vertices of the same community. When all vertices are assigned to the same cluster $HOMOG = 0$ and $COMP = 1$, whereas if each vertex is assigned to a different cluster $HOMOG = 1$ and $COMP = 0$. The Adjusted Random Index (ARI) measures the similarity of the assignments [14]. It ranges from -1 to 1, where $ARI = 1$ corresponds to perfect assignments, ARI values near 0 correspond to bad assignments and negative values of ARI correspond to independent assignments.

3 The community detection algorithm

DAHCA makes use of the reachability matrix and it contains information about the total number of paths between vertices. This was initially proposed in [15] and it was defined as

$$\mathbf{W} = \sum_{l=0}^{\infty} (\alpha \mathbf{A})^l = [\mathbf{I} - \alpha \mathbf{A}]^{-1} \quad (1)$$

where \mathbf{A} is the adjacency matrix and \mathbf{I} is the identity matrix. The parameter α is tuned so that longer paths contribute less and the sum converges. In our case we defined the reachability matrix as

$$\mathbf{A}^l = \sum_{i=1}^l \mathbf{A}^i \quad (2)$$

where every entry \mathbf{a}_{ij}^l represents the exact number of 1-paths, 2-paths, ..., l -paths connecting vertex i with vertex j . We decided to use three-length paths because in many networks, communities overlap. Overlapping vertices serve as bridge between them [16], where most vertices can reach others outside their community in just three hops. Numerical tests also confirmed that it performs best for $l = 3$. Every vertex is then characterised by its relative row entry in the reachability matrix: vertices belonging to the same community will be more likely to have common paths. DAHCA starts by assigning a different cluster to each vertex and a value which consists of the sum of its row entry elements. It then selects the vertex having the lowest value, computes the Euclidean distances between it and all its neighborhoods (non-zero entry elements) and assign it the cluster of the most similar vertex. The process iterates until all vertexes have been assigned to a cluster. Next, the algorithm merges vertices belonging to the same cluster in a new vertex, after which the reachability matrix is recomputed as follow:

$$\mathbf{a}_{ij}^l = \frac{1}{|\mathbf{c}_i|} \sum_{k \in \mathbf{c}_i} \frac{1}{|\mathbf{c}_j|} \sum_{h \in \mathbf{c}_j} \mathbf{a}_{kh} \quad (3)$$

where \mathbf{c}_i and \mathbf{c}_j are the new clusters obtained and \mathbf{a}_{kh} is an element of the adjacency matrix. Figure 1 shows one iteration of DAHCA. At each step a new cluster assignment will be found. The process iterates until change no longer occurs or until one single vertex remains. This can be seen as a degenerate agglomerative hierarchical clustering: each vertex starts with its own cluster and at each iteration clusters are merged until merging is no longer possible. It is different from a classical agglomerative clustering because more than two vertices can be merged together in one iteration and it does not always end with a single cluster including all vertices.

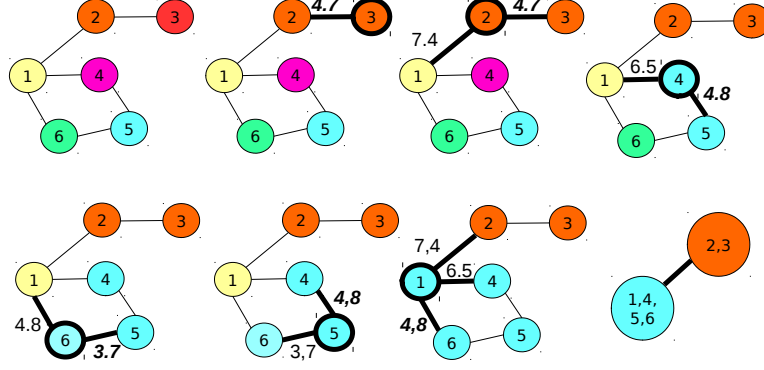


Fig. 1: This figure shows one iteration of DAHCA. A different cluster is initially assigned to each vertex. Vertices are then iteratively selected and they will be assigned the cluster of the closest vertex according to Euclidean distance. Nodes belonging to the same cluster are then merged in a single vertex.

3.1 Complexity

The computational complexity of a community detection algorithm is crucial, especially for large graphs and in cases where the graph is not completely accessible. The complexity analysis of DAHCA can be assessed looking at its phases separately.

- the reachability matrix can be computed in time $O(|V|^l)$ (where $l = 3$ in our case).
- the clustering process can be computed in time $O(|V|)$.
- the merging process can be computed in time $O(|V|^2)$

Notice that the size of V only corresponds to the actual number of vertices during the first iteration. After that, they are merged according to the cluster they belong to, thus the size of V decreases significantly. The overall complexity of DAHCA is then $O(t(|V|^3 + |V|^2 + |V|)) \simeq O(|V|^3)$ where t is the number of iterations. Table 1 shows the time complexity for all algorithms discussed in this paper.

4 Experiments

We have evaluated how effectively DAHCA can detect both communities and the emergence of nested communities. To do so, we have used a similar benchmark as the one described in [16]. Networks have N number of vertices, are divided in G groups and each group is divided in C communities. Vertex connectivity

Table 1: Computational complexity for the different algorithms.

Algorithm	Complexity
BETW	$O(V E ^2)$
GREEDY	$O(V \log^2 V)$
LAB PROP	$O(E)$
TRAP	$O(E V ^2)$
INFOMAP	$O(E)$
EIGEN	$O(E + N)$
DAHCA	$O(V ^3)$

is set to K , this means that each vertex will be connected to exactly K other vertices in the same community. Benchmark graphs have been generated with $N = 120$, $M = 3$, $C = 2$, $K = 5$. Every edge is then relinked with probability p_r . If so, the vertex is connected to another vertex in the same community with probability p_c , in the same group with probability $(1 - p_c)p_g$ or to any vertex in the network with probability $(1 - p_c)(1 - p_g)$. Edges have been relinked with probability $p_r = 1.0$ and $p_g = 0.7$, while p_c was dynamically changed to simulate the emergence of communities (notice that this setting is slightly different from the one presented in [16]). For $p_c = 0$ there is no community structure and only the groups are defined, while for $p_c = 1$ the community structure emerges very clearly. Results are shown in figure 2. For low values of p_c DAHCA is able to identify the correct number of groups, while for higher values it is able to identify the correct number of groups as well as the correct number of communities.

We also evaluated DAHCA on the Girvan-Newman (GN) benchmark [4] and compared it to some state-of-the-art algorithms used for community detection. Networks have N vertices that are assigned to C equally sized communities. Each vertex has a fixed average degree z . A mixing parameter μ controls the portion of intra-community edges. For $\mu = 0$ communities are completely isolated, for $\mu = 0.5$ vertices will be equally connected to vertices inside and outside their community, while for $\mu = 1$ vertices inside the same communities are not connected at all. Benchmark graphs have been generated with $N = 128$, $C = 4$ and $z = 16$, while μ was dynamically changed. The most used metric for community detection is the Normalized Mutual Information (NMI), but it does not necessarily return zero when the assignment is completely random. In that case it depends on the network size and number of communities [14]. This happens when an algorithm assigns each vertex to a different cluster or all vertexes to the same cluster. Therefore we decided to compute completeness and homogeneity to identify when an algorithm returns these naive assignments. For example, the INFOMAP algorithm scores $NMI = 0$ for high values of μ (figure 3a): one cannot say whether it is due to a very bad assignment, a random assignment or just a naive assignment. Also, using homogeneity and completeness it scores $HOMOG = 0$ and $COMP = 1$ (figure 3c and 3d) and clearly assigns every vertex

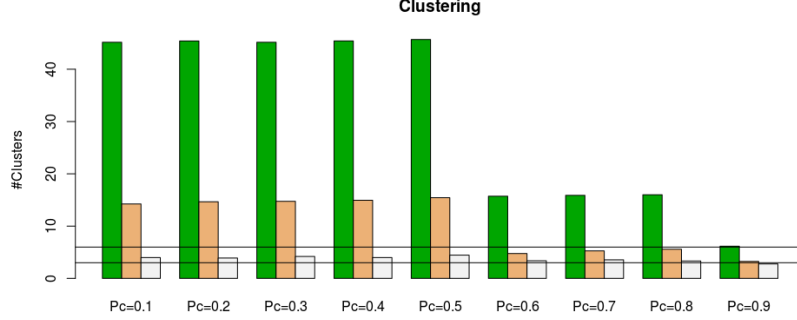


Fig. 2: Clustering over three consecutive iterations. Experiments for $N = 120$, $M = 3$, $C = 2$, $K = 5$, $p_r = 0.2$, $p_g = 0.7$. The probability p_c can be found on the x-axes and the number of clusters identified on the y-axes. Each barplot shows the results obtained for a specific p_c value and each single column represents the number of clusters identified at a certain iteration. The two horizontal lines correspond to the total number of groups (3) and the total number of communities (6) in the networks. Experiments have been run 200 times and results averaged.

to the same cluster. We also decided to use the ARI because, unlike the NMI, it is always independent of the network size and number of communities.

For low values of μ DAHCA does not perform perfectly, unlike some of the other algorithms. However, for $\mu \in [0.3, 0.6]$ it outperforms GREEDY, INFOMAP, LAB PROP and EIGEN. For higher values of μ it outperforms all other algorithms but BETW. Furthermore, it exhibits an interesting behaviour: for $\mu \in [0.75, 1.0]$ there is an increase in performance. One would assume that performance should decrease for $\mu \geq 0.5$ because communities do not become evident, but as proved in [17] they are actually evident for $\mu \leq 0.75$. Over that range, the number of intercommunity edges becomes much higher than the number of intracommunity edges (an "anti-community" structure), with $\mu = 1.0$ being the point where there are no more edges inside communities. DAHCA is able to detect anti-communities which explains why DAHCA's performance increases.

Finally, we have evaluated DAHCA on a real-world network like the Zachary's karate club network, initially presented in [5] and known to be a vastly used benchmark for community detection algorithms. Every vertex represents members of the club, with 1 and 34 being the administrator and the director (the leaders of the two communities). Edges represent friendship between members. Results for all algorithms are presented in figure 4 and 5, while the result obtained using DAHCA is presented in figure 6. Nodes belonging to the same communities share the same colour. The algorithms obtain very different results, especially for the number of communities found. Firstly, all algorithms are able to identify vertices 1 and 34 as community leaders and assign them

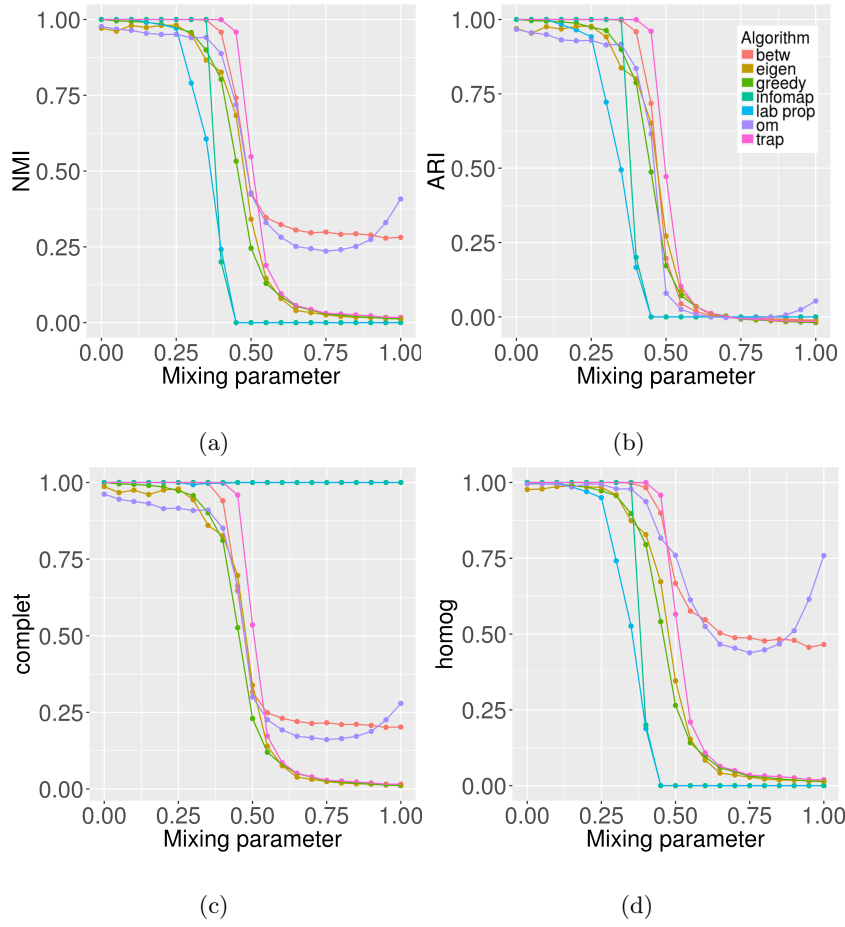


Fig. 3: Experiments on GN benchmark for $N = 128$, $C = 4$, $K = 16$. The Mixing parameter μ can be found on the x-axes and NMI, ARI, completeness and homogeneity on the y-axes. Experiments have been run 50 times and results averaged.

to different communities. Only DAHCA and LAB PROP are able to identify the core sets of vertices $\{1,2,3\}$ and $\{33,44\}$ as nodes with higher connectivity [4] and assign them to different communities. To complete the analysis, table 2 shows the numerical results of the different algorithms on the Zachary network.

Table 2: Metrics for the different algorithms on the Zachary karate club network

	BETW	GREEDY	LAB PROP	TRAP	INFOMAP	EIGEN	DAHCA
NMI	0.579	0.692	0.548	0.504	0.699	0.677	0.512
ARI	0.469	0.680	0.467	0.333	0.702	0.512	0.452
V measure	0.580	0.692	0.548	0.504	0.699	0.677	0.512
Completeness	0.431	0.577	0.428	0.364	0.593	0.512	0.387
Homogeneity	0.885	0.866	0.763	0.822	0.854	1	0.756

5 Conclusions

In this paper we have proposed a degenerate agglomerative hierarchical clustering algorithm that makes use of the reachability matrix to detect community structures in networks and runs in $O(|V|^3)$. We have tested DAHCA on different benchmarks and settings. First, it has been tested on the benchmark used in [16] where networks are organised in groups and each group is organised in communities. We have shown that it is able to identify both group and community structure. Next we have compared it to some state-of-the-art algorithms on the Girvan-Newman benchmark [4] and discovered that, even if it does not show optimal results on the simplest networks, it is able to outperform most of the other algorithms for the more complex ones. Finally, we have demonstrated the results obtained on the Zachary’s karate club network and discovered that it is able to assign the two community representatives and core members to different communities.

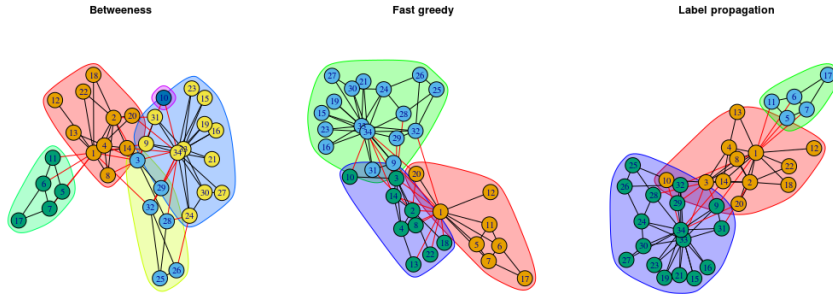


Fig. 4: Communities found by BETW, GREEDY and LAB on the Zachary's karate club network

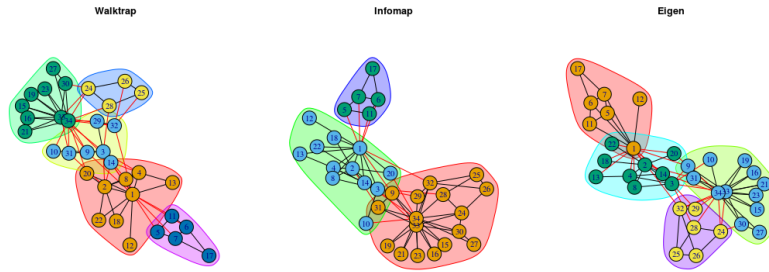


Fig. 5: Communities found by by TRAP, INFOMAP and EIGEN on the Zachary's karate club network.

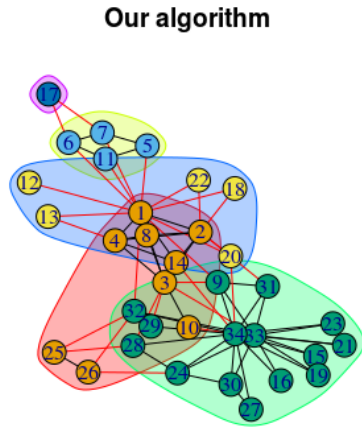


Fig. 6: Communities found by DAHCA on the Zachary's karate club network.

Bibliography

- [1] Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Volume 8. Cambridge university press (1994)
- [2] Albert, R., Jeong, H., Barabási, A.L.: Internet: Diameter of the world-wide web. *nature* **401**(6749) (1999) 130–131
- [3] Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabási, A.L.: The large-scale organization of metabolic networks. *Nature* **407**(6804) (2000) 651–654
- [4] Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proceedings of the national academy of sciences* **99**(12) (2002) 7821–7826
- [5] Zachary, W.W.: An information flow model for conflict and fission in small groups. *Journal of anthropological research* **33**(4) (1977) 452–473
- [6] Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Physical review E* **69**(2) (2004) 026113
- [7] Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. *Physical review E* **70**(6) (2004) 066111
- [8] Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Physical review E* **76**(3) (2007) 036106
- [9] Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: *ISCIS*. Volume 3733. (2005) 284–293
- [10] Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* **105**(4) (2008) 1118–1123
- [11] Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Physical review E* **74**(3) (2006) 036104
- [12] Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* **2005**(09) (2005) P09008
- [13] Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. *Scientific reports* **6** (2016) 30750
- [14] scikit learn: User guide - clustering. <http://scikit-learn.org/stable/modules/clustering.html> (2017) [Online; accessed September 25, 2017].
- [15] Katz, L.: A new status index derived from sociometric analysis. *Psychometrika* **18**(1) (1953) 39–43
- [16] Bagnoli, F., Massaro, E., Guazzini, A.: Community-detection cellular automata with local and long-range connectivity. *Cellular Automata* (2012) 204–213
- [17] Lancichinetti, A., Fortunato, S.: Erratum: Community detection algorithms: A comparative analysis [*phys. rev. e* 80, 056117 (2009)]. *Physical Review E* **89**(4) (2014) 049902